IBM Software Group

Agile & Secure Software Development

A vintage year? Or just old wine in a new bottle?

Andras R. Szakal,
IBM Distinguished Engineer
Director Software Architecture, IBM Federal
aszakal@us.ibm.com

Go to IBM

Agenda

- The Business Case for Agility
- Agile Development
- IBM's Internal Agile Strategy
- IBM Rational Agile Solutions
- Static Code Analysis within Agile Development using Jazz
- Conclusion

Today's reality: An intense focus on business outcomes

Align *IT* investments with rapidly evolving business priorities

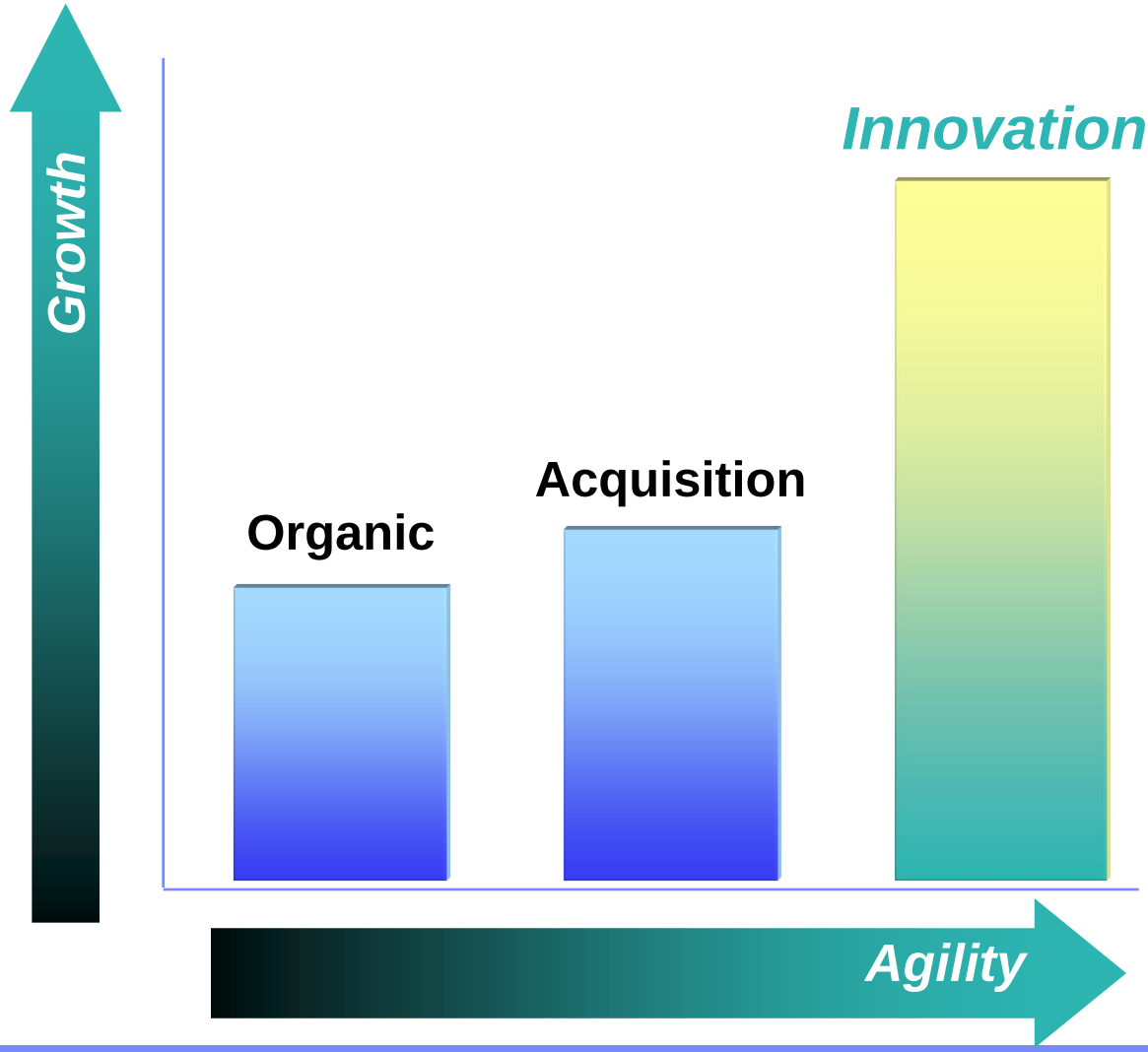
Manage value and mitigate risks by improving project management

Control costs & improve global operational efficiencies

Ensure security and compliance in a changing global environment

Integrate value in organizationally diverse environments

Agility is Essential to any Growth Strategy…



…and companies that demonstrate the highest levels of Agility and Innovation realize the greatest growth

What Does an Agile Organization Look Like?

Characteristics	Behaviors
Deep, continuous **alignment** between business and IT	Active LOB participation in IT strategy formation
Interdisciplinary **skills**	IT is part of business strategy, not an afterthought
Heavy internal and external use of **business technology**	IT is key to productivity, collaboration, channel, and decision-making
Governance that makes it easy to make the right decision	Reuse is incentivized; clear decision rights management; clear, enforced policies
IT's strategic importance reflected in org chart	CIO is part of executive council

Creates pressure on software delivery and solution sourcing


"2/3 of our projects span multiple business units and locations"


"People knew it was a train wreck; no one could see what to do"


"The business is always changing what they want and we are the last to know"


We run on the Web… we run fast so new code makes us really nervous"


"As-is won't do, we need new business models and capture the best talent"

- Increasing reliance on out sourced partnerships & software reuse
- Interest in replicating the proven models of open, community-governed software delivery models
- Popularity of Agile development
- Investments in SOA and Web 2.0 to enable new business models

Impact of growing modularity

- More granular service functionality used across business applications
- Larger number of projects and assets including custom, outsourced and packaged

Impact of accelerating change

- Asset are impacted by frequent updates and changing interdependencies
- Effective cross-organizational visibility and synchronization becomes an imperative!

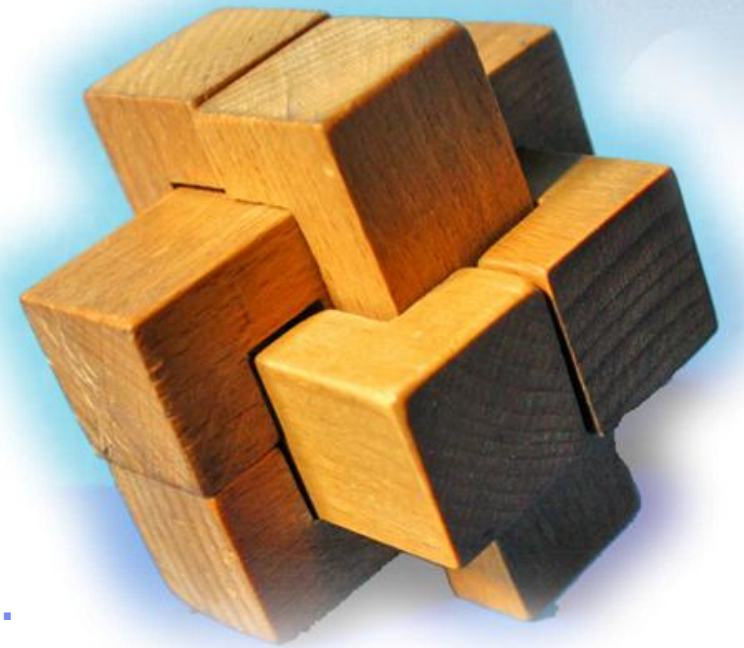
Agenda

- The Business Case for Agility
- Agile Development
- IBM's Internal Agile Strategy
- IBM Rational Agile Solutions
- Static Code Analysis within Agile Development using Jazz
- Conclusion

What is Agile?

- An iterative and incremental (evolutionary) approach performed in a highly collaborative manner with **just the right amount of ceremony** to produce high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.

- Core principles
  - “Fits just right” process
  - Continuous testing and validation
  - Consistent team collaboration
  - Rapid response to change
  - Ongoing customer involvement
  - Frequent delivery of working software



...focus on predictability v.s. repeatability...

Agile goes Mainstream

From the analyst community

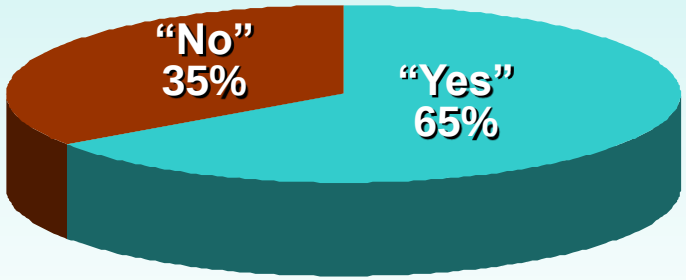
"Thirty-five percent of ….respondents have projects or pilots underway, and *only 12 percent do not see a fit* for Agile processes in their organizations.

The fact that 88 percent of these organizations (one-third of which have over 10,000 employees) are using or evaluating Agile processes proves that Agile processes have truly hit the mainstream."

- Excerpt from "And the Agile Survey Says…" Agile Journal, March 6, 2006

Third-party research suggests even wider adoption

Have you adopted any Agile techniques?



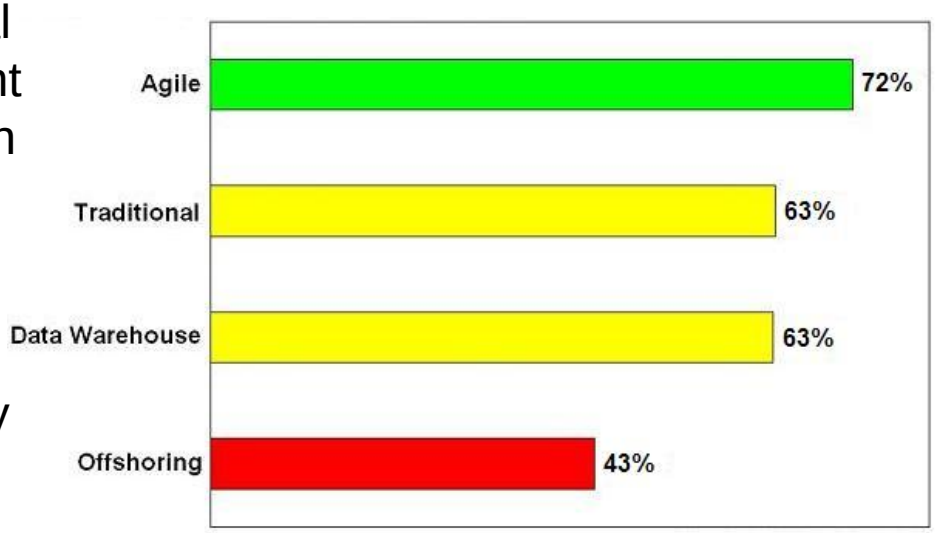
Source: 'Agile Adoption Rate Survey' of over 4200 Dr. Dobb's subscribers, March 2006

18.3% of respondents indicated they're still in the pilot stage

15% of "No" respondents hope to do Agile this year

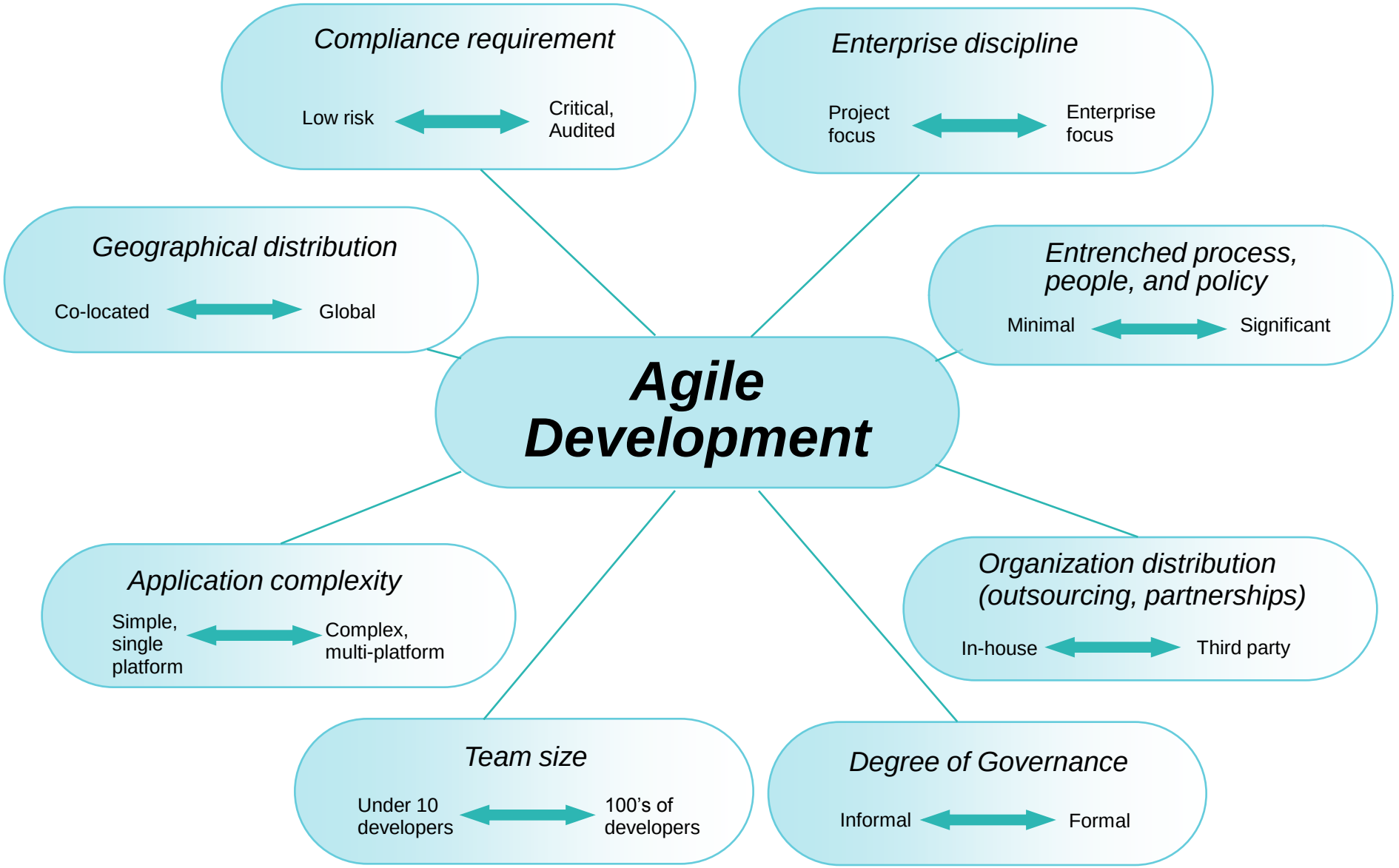
Why Agile/Lean? It's More Successful

- Quality: 87% believe that delivering high quality is more important than delivering on time and on budget
- Scope: 87% believe that meeting actual needs of stakeholders is more important than building the system to specification
- Money: 80% believe that providing the best ROI is more important than delivering under budget
- Staff: 76% believe that having a healthy workplace is more important than delivering on time and on budget
- Schedule: 62% believe that delivering when the system is ready to be shipped is more important than delivering on schedule

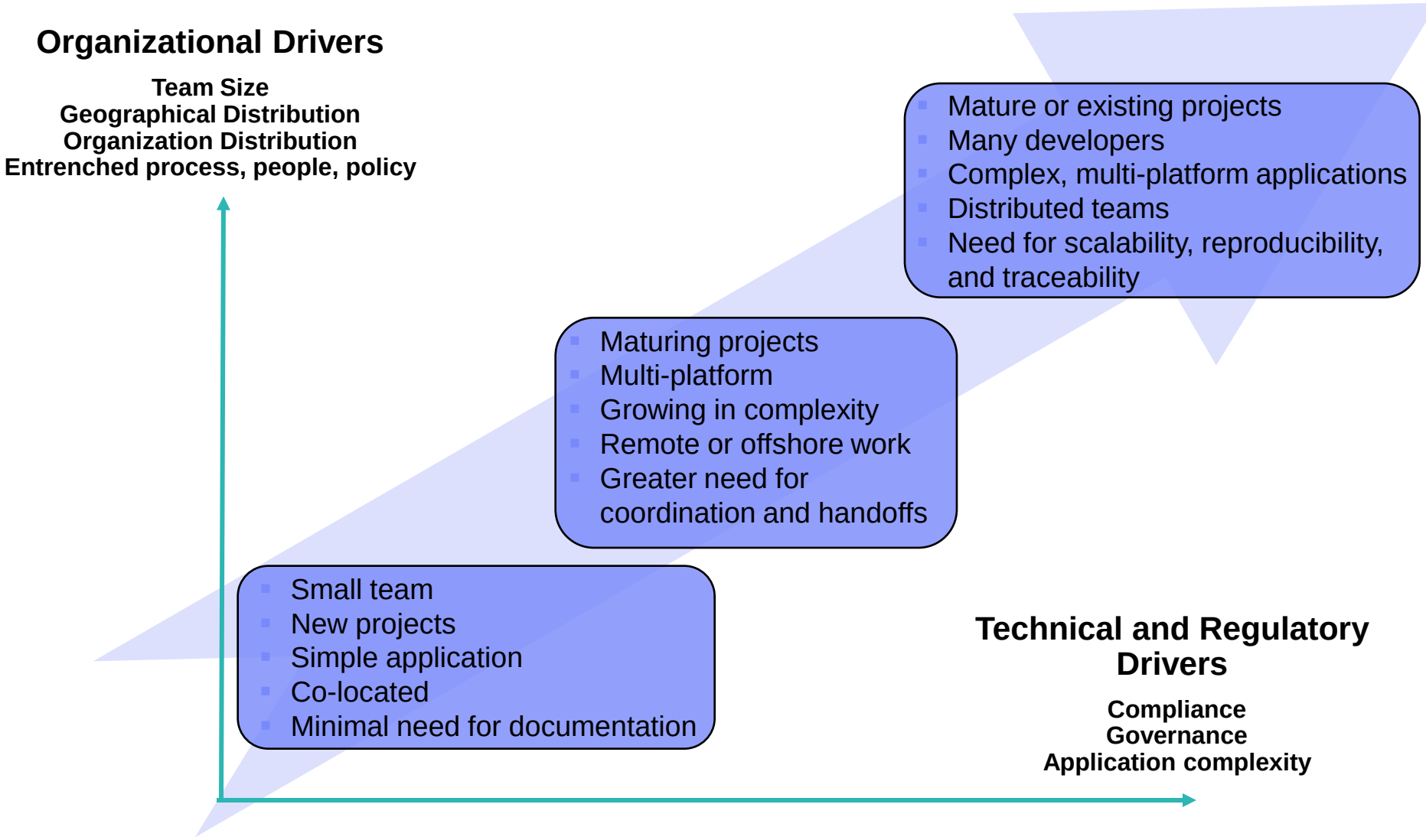


Source: Dr Dobb's 2007 Project Success Survey

Challenges with Agile in the Mainstream



Agility is Relative – It Depends on Project Dynamics



Agenda

- The Business Case for Agility
- Agile Development
- IBM's Internal Agile Strategy
- IBM Rational Agile Solutions
- Static Code Analysis within Agile Development using Jazz
- Conclusion

IBM Agile Development

The view from within...

"Use continuous stakeholder feedback to deliver high quality consumable code through use cases and a series of short, stable, time-boxed iterations."

Agile Manifesto

- Individuals and interactions **over** processes and tools
- Working software **over** comprehensive documentation
- Customer collaboration **over** contract negotiation
- Responding to change **over** following a plan

getting Agile@IBM

Key concept: Agile development necessitates *greater* discipline than traditional methods. "Quality" and "Consumability" must be real, not platitudes.

IBM Agile Principles – An Agile Project:

1. **Leverages the inevitability of change and generates new occasions for learning throughout the project.**
2. **Provides leadership to create an atmosphere in which the team determines its capacity and commits to shared goals.**
3. **Uses frequent interaction to move the whole product team toward its goals.**
4. **Engages with customers and other stakeholders throughout the project to generate continuous feedback.**
5. **Measures success in terms of delivering a flow of functional, proven stakeholder-valued capabilities.**
6. **Employs test-driven development and Agile Model Driven Development and is intolerant of defects.**
7. **Strives for relentless improvement of the product and the process.**

IBM Development Community - Leveraging Collaboration

Government Standards – Enablement Artifacts

Collaborate:

1. One-on-one consultations
2. Process improvements to remove obstacles
3. Presentations and group discussions

Statistics:

Tool/Reusable Code:
-IPv6 Compliance Wizard: 168 users
-IPv6 Hostname Validator & Test Tool: 76 downloads

Artifacts from wiki (as of April 2008):
-IPv6 Related: 551 downloads
-FIPS Related: 873 downloads
-Common Criteria Related: 800 downloads

TOTAL: 2224 wiki downloads

Educate:

1. IPv6 Implementation Requirements
2. FIPS 140 Guideline
3. [redacted]
4. Common Criteria Cookbook
5. Lessons Learned from Tivoli License Compliance Manager CC Evaluation
6. Lessons Learned from WebSphere Federation Server CC Evaluation
7. What is the "+" in EAL4+ and What Needs To Be Done to Get It
8. Common Criteria Starter Kit
9. [redacted] CC Self Assessment Form
10. [redacted] CC Checklist for EAL2
11. [redacted] CC Checklist for EAL3
12. [redacted] CC Checklist for EAL4
13. [redacted] Initial Questionnaire
14. [redacted] Initial Assessment Agenda

Enable:

1. IPv6 Compliance Wizard
2. IPv6 Hostname Validator
3. IPv6 HTTP Stress Test Tool

Gov Stds community reuse / collaboration growing

Enable: increased reuse

	Oct 2007 (cumulative)	Oct 2008 (cumulative)	Increase Percentage
IPv6 Wizard usages	135	214	58%
IPv6 Reusable code download	48	68	42%

"… these resources save plenty of our time and provide a consistent source of all necessary information that individual development groups can use to achieve IPv6 enablement of their products "

Tivoli Network Management Development Team

Collaborate:

- **Resolve dependency issues to enable deliveries**
- **One-on-one consultations with product teams, resulting in mitigations of several revenue-impacting issues**

Educate: increased wiki traffic and artifact download

	Oct 2007 (cumulative)	Oct 2008 (cumulative)	Increase Percentage
Wiki visitors	3680	9198	150 %
Artifact download	1547	3529	128 %

IBM Internal Agile Success: Tivoli Workload Scheduler

"By developing our product in iterations and setting checkpoints at least monthly for all the development activities, we were able to make necessary changes to the architecture and avoid the pain of discovering such problems in the later phases of the development cycle."

Environment

- Thousands of customers
- ~70 team members
- IBM Rational Quality Management products
- IBM Rational Method Composer and Rational Unified Process

Agile Techniques Used

- Use case based development approach
- Frequent iterations
- Continuous testing
- Ongoing customer involvement
- RUP for process improvement

Results

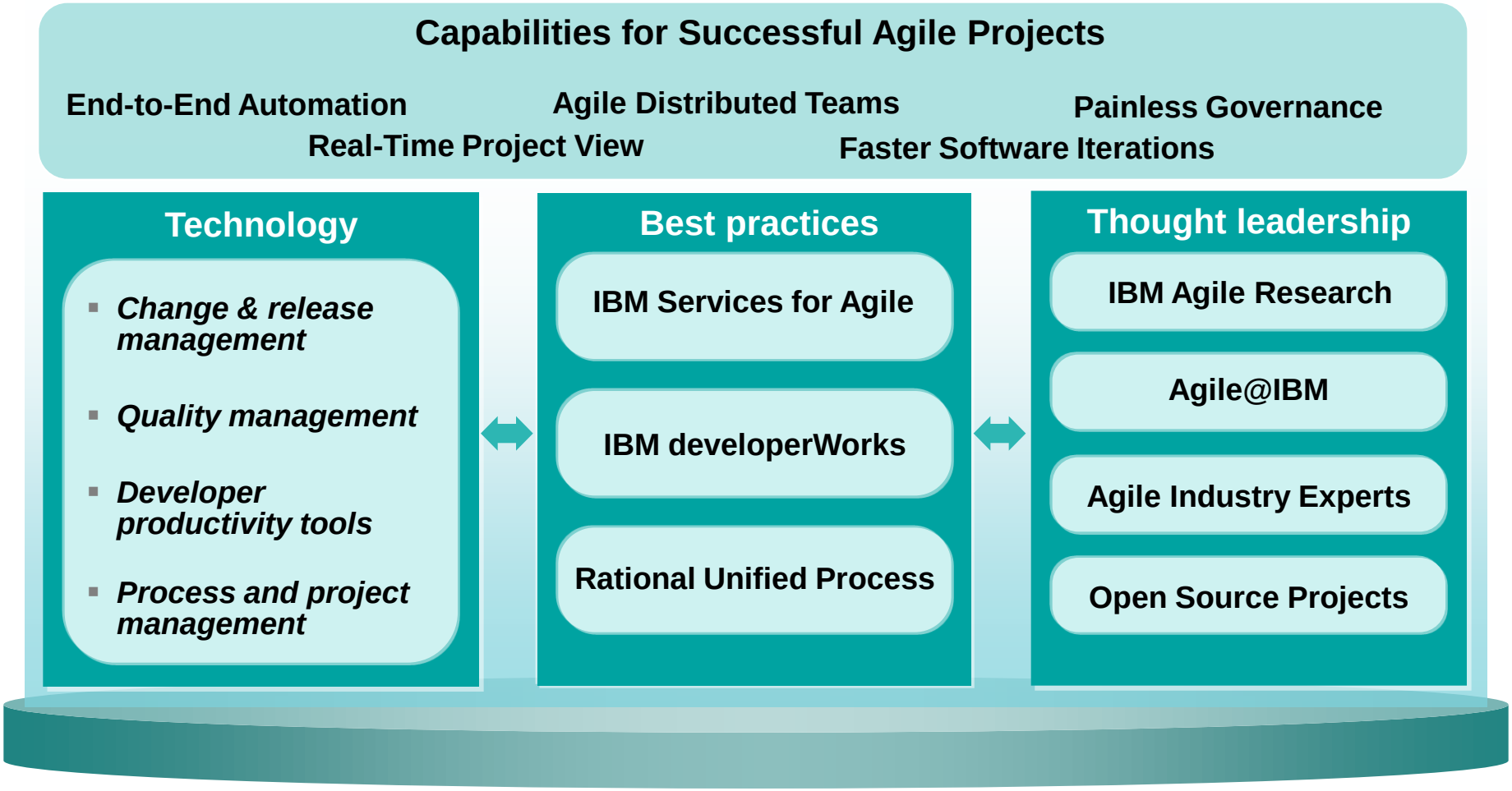
- 30% increase in developer productivity
- Design level defects cut in half
- QA productivity increased by 20%
- Test coverage increased by 30%
- 200% ROI in first release

Agenda

- The Business Case for Agility
- Agile Development
- IBM's Internal Agile Strategy
- IBM Rational Agile Solutions
- Static Code Analysis within Agile Development using Jazz
- Conclusion

IBM Rational Solution for Agile Development

Enabling rapid global software delivery



The Jazz project

People working together to deliver great software

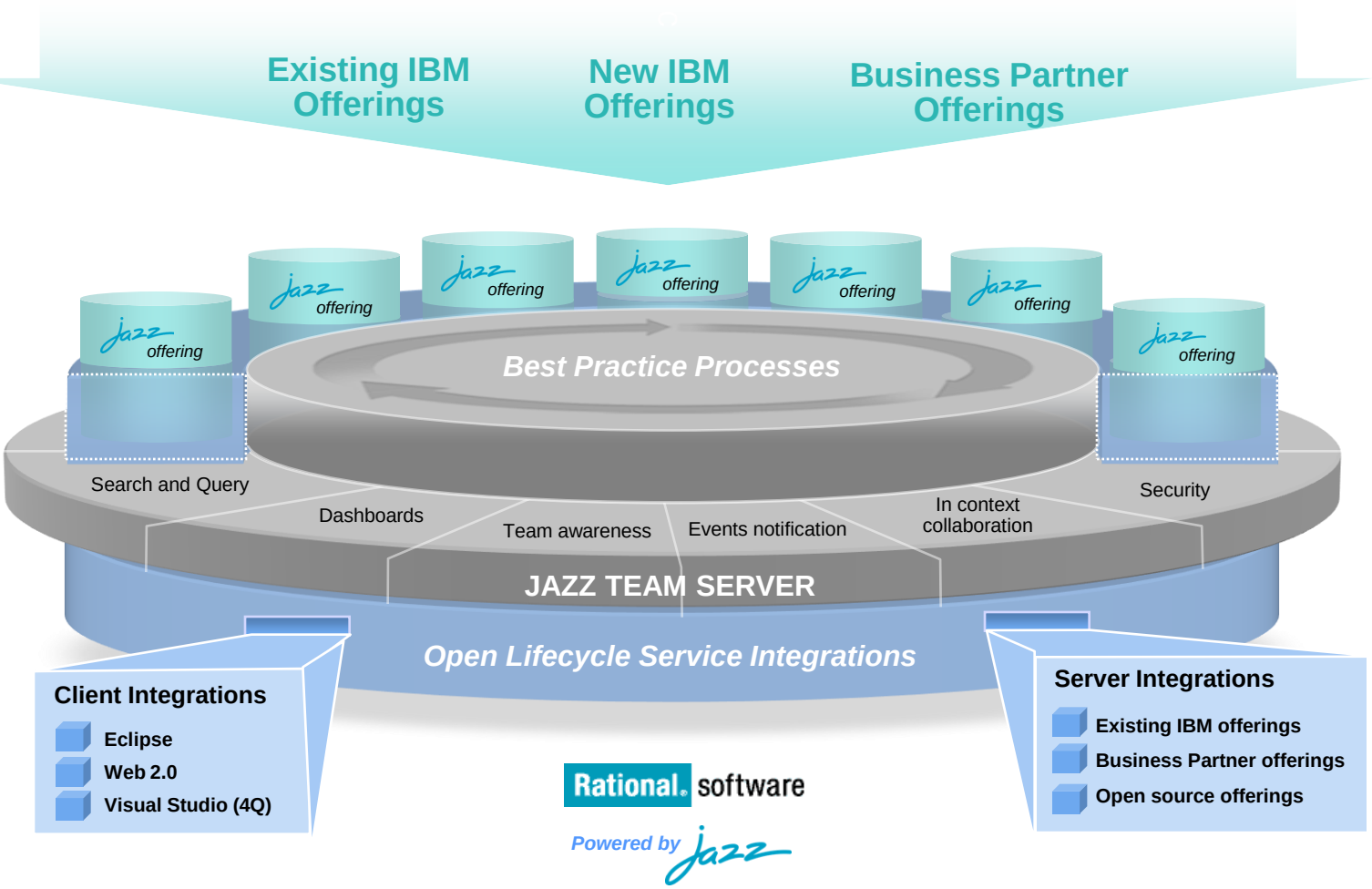
Jazz is a project and technology for *transforming how people work together* to deliver greater value and performance from their software investments.

jazz

- A major investment by IBM to create **a scalable, extensible team collaboration foundation**.
- IBM's vision of the **future of software delivery**–globally distributed, fluid and dynamic.
- An **evolution of the Rational portfolio**, which will evolve to support Jazz technology over time.
- **A community at Jazz.net** – where you can see Jazz-based products being built.

21

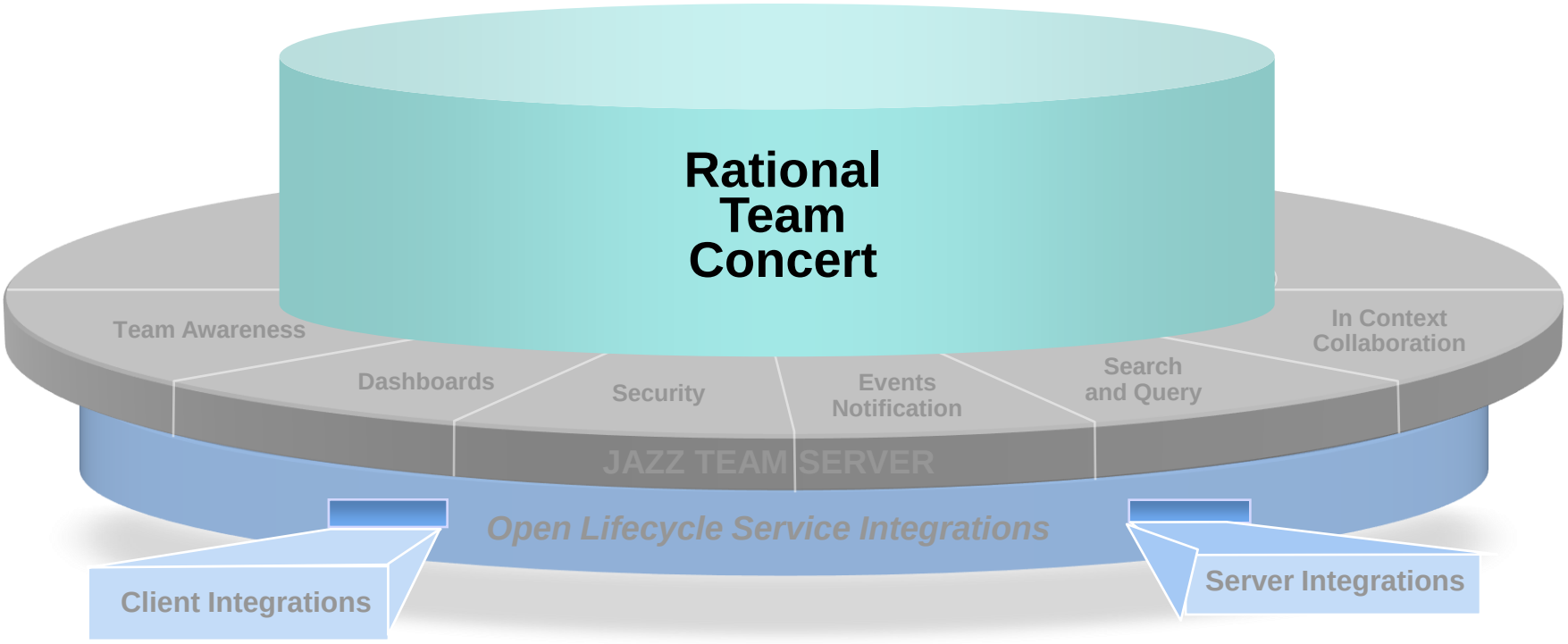
An evolution of value in collaborative ALM bringing forward existing investments and delivering new innovative capabilities



22

Today's focus: Rational Team Concert

The first offering built on Jazz technology



Rational. software

Software innovation through collaboration

- **Real time, in-context team collaboration**
  - Make software development more automated, transparent and predictive
- **"Think and work in unison"**
  - Integrated source control, work item and build management
- **Assess real-time project health**
  - Capture data automatically and unobtrusively
- **Automate best practices**
  - Dynamic processes accelerate team workflow
  - Out-of-the-box or custom processes
- **Unify software teams**
  - Integrate a broad array of tools and clients
  - Extend the value of ClearQuest and ClearCase
  - Visual Studio Client (1Q09)
  - Support for System z and System i servers (Q4)

Open and extensible on Jazz

- ✓ Collaborate in context
- ✓ Right-size governance
- ✓ Day one productivity

Solutions for Agile Teams of Every Size

Rational Team Concert (Jazz)
Rational AppScan
Rational Application Developer
EPF / OpenUp
Rational Build Forge
Rational ClearQuest
Rational PurifyPlus

Small Agile Teams
(under 10 developers)

New projects
- Simple application
- Co-located
- Minimal need for documentation

Rational Team Concert (Jazz)
Rational AppScan
Rational Method Composer
Rational Application Developer
Rational ClearCase
Rational Build Forge
Rational ClearQuest
Rational PurifyPlus
Rational Performance, Functional, and Manual Tester
RUP for Large & Distributed Projects

Large Scale
Agile Teams

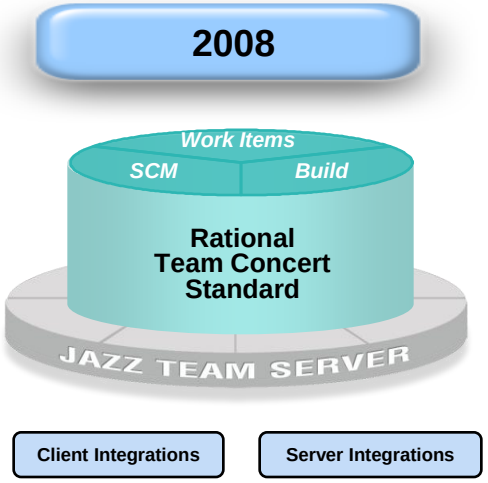
Mature or existing projects
- Many developers
- Complex, multi-platform applications
- Distributed teams
- Need for scalability, reproducibility, and traceability

Mid-sized Agile Teams

Maturing projects
- Multi-platform
- Growing in complexity
- Remote or offshore work
- Greater need for coordination and handoffs

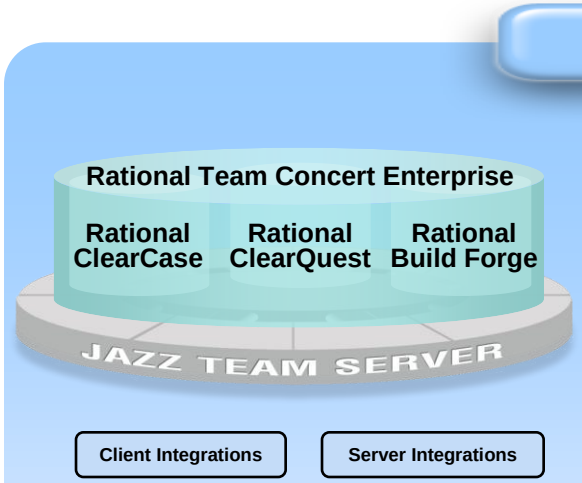
Rational Team Concert (Jazz)
Rational AppScan
Rational Application Developer
Rational Build Forge
Rational ClearCase LT
Rational ClearQuest
Rational PurifyPlus
Rational Performance Tester
RUP for Small Projects

Statement of direction: Rational Team Concert



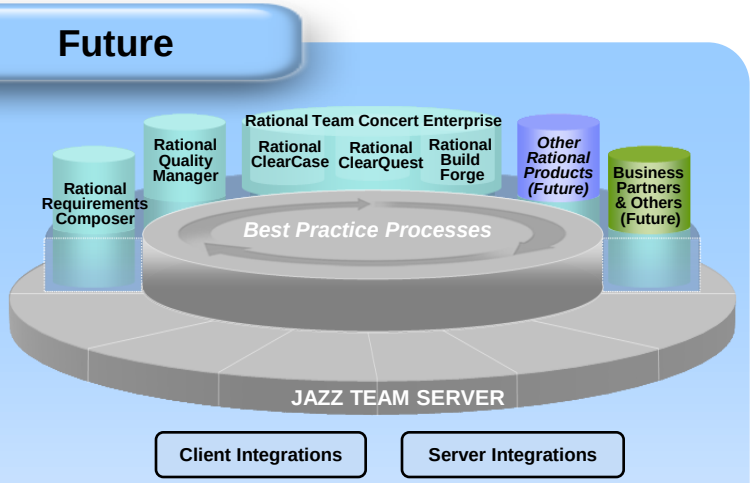
Rational Team Concert Standard

- A collaborative development environment optimized for small and medium, distributed teams
- All the capabilities of the Jazz platform
  - Plus work item, SCM and build management
- Integrates with Rational Build Forge for enterprise build
- Interoperates with Rational ClearCase and ClearQuest
- Subversion connector available



Rational Team Concert Enterprise

- A collaborative development environment for any size team
- All the capabilities of the Jazz platform
  - Plus full future versions of Rational ClearCase, ClearQuest and Build Forge
- Optional connectors provide seamless interoperation with enterprise repositories



Rational Software Delivery Platform 2.0

- Comprehensive collaborative application lifecycle management
- Frictionless software delivery spanning
  - Change and release management
  - Requirements definition
  - Test management
  - Enterprise reporting
  - More
- Open, extensible architecture expands tooling options

All capabilities in lower version are available in higher version.

**Statements on future direction subject to change*

Agenda

- The Business Case for Agility
- Agile Development
- IBM's Internal Agile Strategy
- IBM Rational Agile Solutions
- Static Code Analysis within Agile Development using Jazz
- Conclusion

RSAR Integration with RTC (Jazz)

- **Rational Static Code Analyzer** (RSAR) integrates directly into RTC
- Provides extensibile framework for static code analysis
  - Unify your tools: one scan, one view, one analysis, one decision
  - Execute multiple scan rules and tools from a common framework
  - Central management of legacy and third-party technologies
- Integrates with leading analysis tools
- Can employ rules against more than one provider
- Enforces coding standards against more than just common vulnerabilities – examples, Globalization, accessibility, etc.

IBM Rational Software Analyzer

Extensible framework: Customizable to meet your specific needs

- Multiple Application Programming Interface (API) extension points provide the ability to:
  - Create your own language parsers for custom code analysis
  - Create groups of rules to address specific business needs
  - Create custom rules and analysis views to enforce company best practices and policies
  - Create your own custom reports and views to see results in your preferred format
  - Create new data collection points to collect the data you want
  - Enforces RTC / Jazz Agile Development Rules

Establishing an Agile Secure Engineering Community

- Teams need to collaborate across the development lifecycle
- Assurance practices are dynamic and change frequently
- Framework Based
- Automated

IBM's Secure Engineering Framework

- Enabled Through a Secure Engineering Framework
  - Decision Support Matrix
  - Best Practices
- Defines assurance activities by phase by product type
  - Set of deliverables / artifacts appropriate for the product profile (architecture)
- Establishes Risk Management Framework for Secure Engineering
  - Identify implications of decisions
  - Establishes mechanism for Risk Management / Mitigation

Leveraging Web2.0 to Facilitate an SE Community

Secure Engineering Community

Make this my preferred communi
Join the communit

Over the last few years, IBM and its customers have seen a disturbing rise in the number of successful attempts to leverage IT product vulnerabilities to obtain personal, corporate and government data by organized crime and even rogue member states. Recently, concerned customers have asked IBM for more details on how we ensure the quality and integrity of our software and hardware offerings; in particular, how IBM develops secure software/hardware -- how we ensure that our products are secure, or securable, and are unlikely to contain security vulnerabilities.

In addition, the U.S. federal government is considering policies which would limit the acquisition of IT products that do not meet certain assurance levels and use of secure coding standards by the vendor. In order to preserve our competitive advantage, we must continue to focus on these changing business dynamics and the resulting impact to our secure engineering practices.

The mission of the secure engineering community is to:

- View community information on the Secure Engineering Community Wiki
- Ask questions and provide answers in the Secure Engineering Discussion Forum
- View this community on the Lotus Community Server

Wiki's

Forums

Feeds

Tagging

Asset Management

Community bookmarks

- Secure Engineering Wiki
- Secure Engineering Forum

Community feeds

Chris DeRobertis' Data Alchemy @ BlogCentral

Expertise

Subject Matter Experts

Andras Szakal
Click here for Bluecard
SWG

Christopher V Derobertis
HPC Cluster Security Architect/Team Lead

Timothy Hahn
Distinguished Engineer, Chief Architect, Secure Systems and Networks

Tony Nadalin
Distinguished Engineer, Chief Security Architect

Brian Snitzer
Senior Technical Staff Member

Nikola Vouk
ITM Agent Developer (currently UNIX OS Agent)

Jim Whitmore
Senior Certified IT Architect

Welcome Andras R. Szakal | Log Out | Refresh BGs
SECURE ENGINEERING COMMUNITY
Home
IBM Secure Engineering
Search WCv2
GO
Searching seceng

WikiCentral v2 Home
Dashboard
Secure Engineering Community
News
Home

Dashboard > Secure Engineering Community > Home

Published on Apr 29, 2008

About Secure Engineering Community

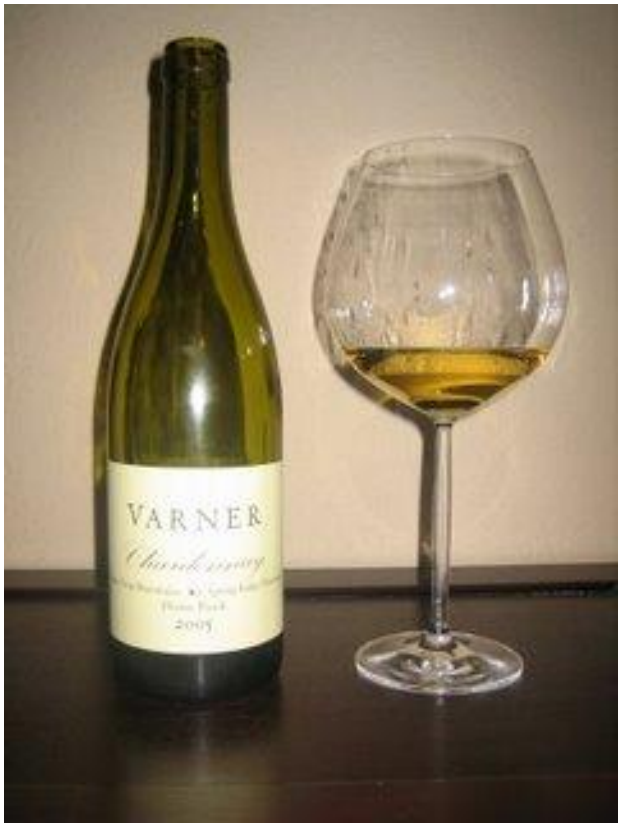
IBM has long been considered a strong producer of high quality hardware and software. This reputation is well founded since many of our products are used in high performance, high stress, highly available environments which many of our customers view as mission critical to their businesses. Our customers and indeed many of our competitors look to us to lead the way in defining development practices

Secure Engineering Community - Leveraging Collaboration

- Development Automation is Key to Developer Adoption
  - Low ceremony = High Rate of Adoption
- Framework must be tools based
- Tooling must facilitate the development of the community
  - Collaborative team focused
  - Leverage Web2.0 Tooling
  - Integrate Education Services

Conclusion

- Agile adoption must be
  - Collaborative
  - Automated
  - Empowered
  - Flexible
  - Adaptive



...then and only then will have fine wine...

धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

شكرا

Arabic

Thank You

Obrigado

Brazilian Portuguese

Grazie

Italian

Danke

German

Merci

French

நன்றி

Tamil

多谢

Simplified Chinese

감사합니다

Korean

ありがとうございました

Japanese

Introducing IBM Rational Team Concert

Software innovation through collaboration

- Think and work in unison
  - Unify functional, organizational and geographic silos
  - Integrated source control, work item, and build management built on the Jazz team server
- Streamline everyday activities
  - Dynamic processes automate hand-offs
  - Leverage out-of-the-box or custom processes
- Simplify governance
  - Automated data gathering
  - Transparency of teams and processes
  - Real-time project health information
- Unify teams that use a variety of clients
  - Leverage Web, Eclipse or any Eclipse-based product
  - Extend the value of ClearQuest & ClearCase in enterprise deployments

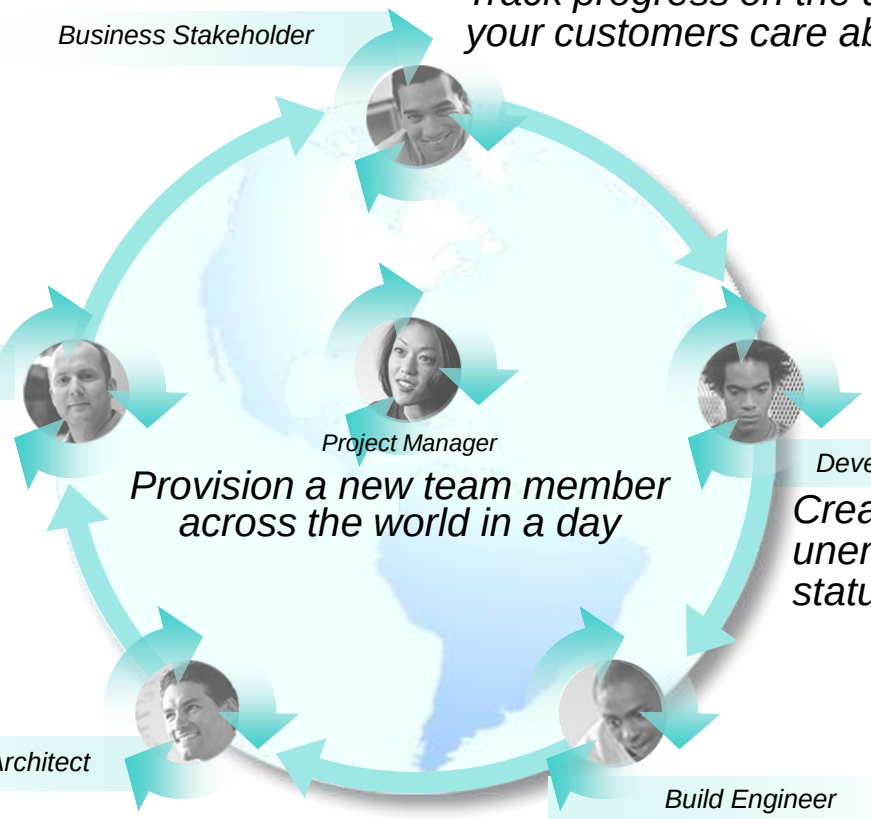
Open and extensible on

jazz

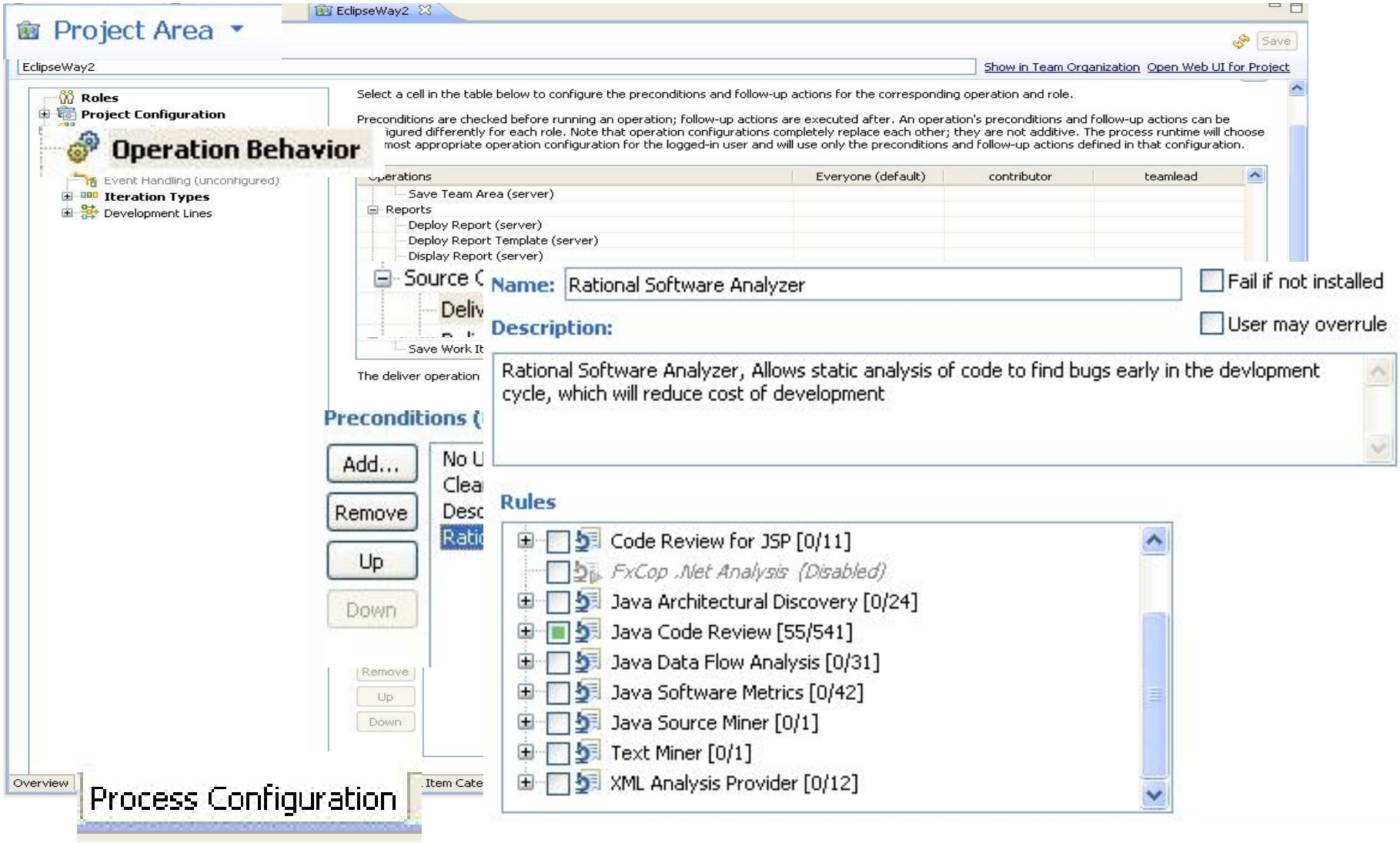
- ✓ Collaborate in context
- ✓ Right-size governance
- ✓ Day one productivity

Delivering global team collaboration and visibility

Allows individuals to focus on what they do best



RSAR and RTC Integration Demo



Delivering customer value with Rational Team Concert

The evolution of software delivery

Collaborate in context

- Fewer meetings and status reports
- 50% reduction in late scrap and rework
- Reduced attrition of skills and greater flexibility to leverage top technical and business talent

Right-size governance

- 50% less overhead to stay compliant with standards
- Earlier detection and accelerated resolution of defects
- Improved predictability of project outcomes

Day one productivity

- Project initiation in days rather than weeks
- Reduction in time to first demonstrable release
- Improved cycle times through automated and reusable patterns, services and architectures

Supporting the software economics of an increasingly global, dynamic and web-based technology environment

IBM Lab Management Philosophy

- **Globally Integrated Development Team**
  - Common architecture, tools and process
  - Lightweight governance
  - Distributed skill pools leveraged world-wide as well as in local engagements
- **Continuous focus on effectiveness and efficiency**
  - Culture of reuse facilitated through common components and architecture
  - Incremental development approach with continuous customer feedback
  - Strong ongoing focus on consumability and quality
- **Strong sustained improvement**
  - Faster delivery, faster feedback, faster deployment, faster renewal rate
  - Reduced base development expense

The defining challenges in effective software delivery

Only 37% of stakeholders are satisfied with the speed of internal application development[1]..

Only 42% are satisfied with the quality[1].

50% of outsourced projects are expected to underperform[2].

Silos of people, process, and projects

Geographic Barriers

- Poor communication
- Language, culture, time
- Process gaps resulting in rework
- High degree of friction

Organizational Barriers

- Lack of meaningful collaboration
- Weak project governance
- Lack of domain expertise
- Poor LOB oversight
- Security of IP when outsourcing

Infrastructure Barriers

- Incompatible tools / repositories
- Unreliable access artifacts
- Lengthy on-boarding
- Inflexible tooling integration

Source: [1]Forrester, [2]Gartner

Customer Speak: What they would like to see possible

Project status and cost analysis anytime for all business sponsors

Top down and the low level views linking configuration, bug and cost tracking

Easily extend and automate short iteration cycles to meet hot business needs

Agile work flows automating and smoothing execution

Empower team members to collaborate using up to date information "in context" of the work they are doing

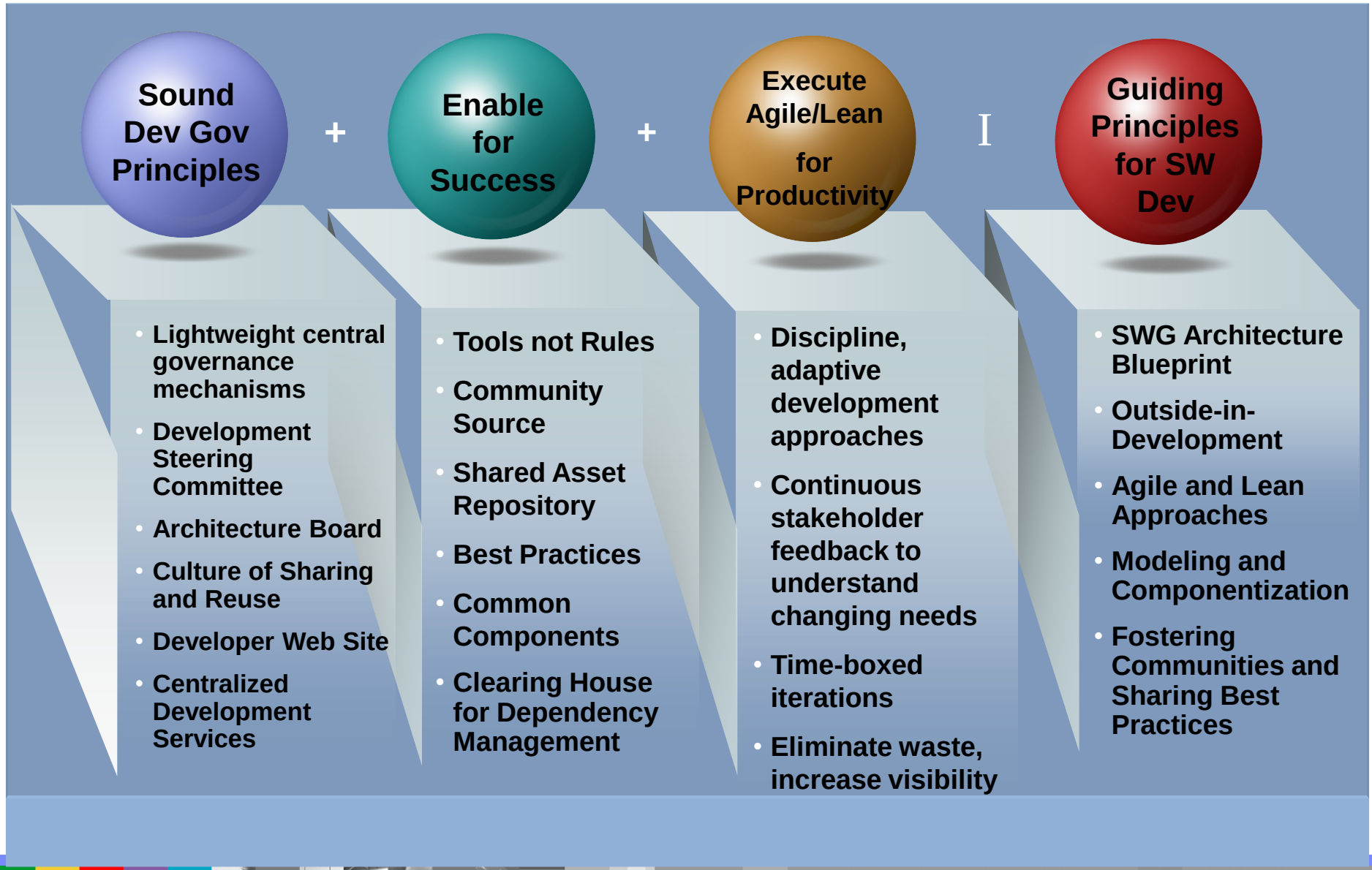
Automated synchronization of project artifacts

Easy triggers to catch licensing issues early

Seamless access to highest risks and security vulnerabilities

Onboard team members across the world across projects in a day not months

Best Practices for Distributed Development Success



Envisioning a technology…

That can transform software delivery

An open technology initiative to *transform how people work together* to deliver greater value and performance from their software investments

- Robust, extensible and scaleable
- Globally distributed, fluid and dynamic
- Community-based and open at Jazz.net

Collaborate in Context

- Enable team transparency of “who, what, when, why”
- Build team cohesion and presence
- Automate hand-offs so nothing falls through the cracks

Right-size Governance

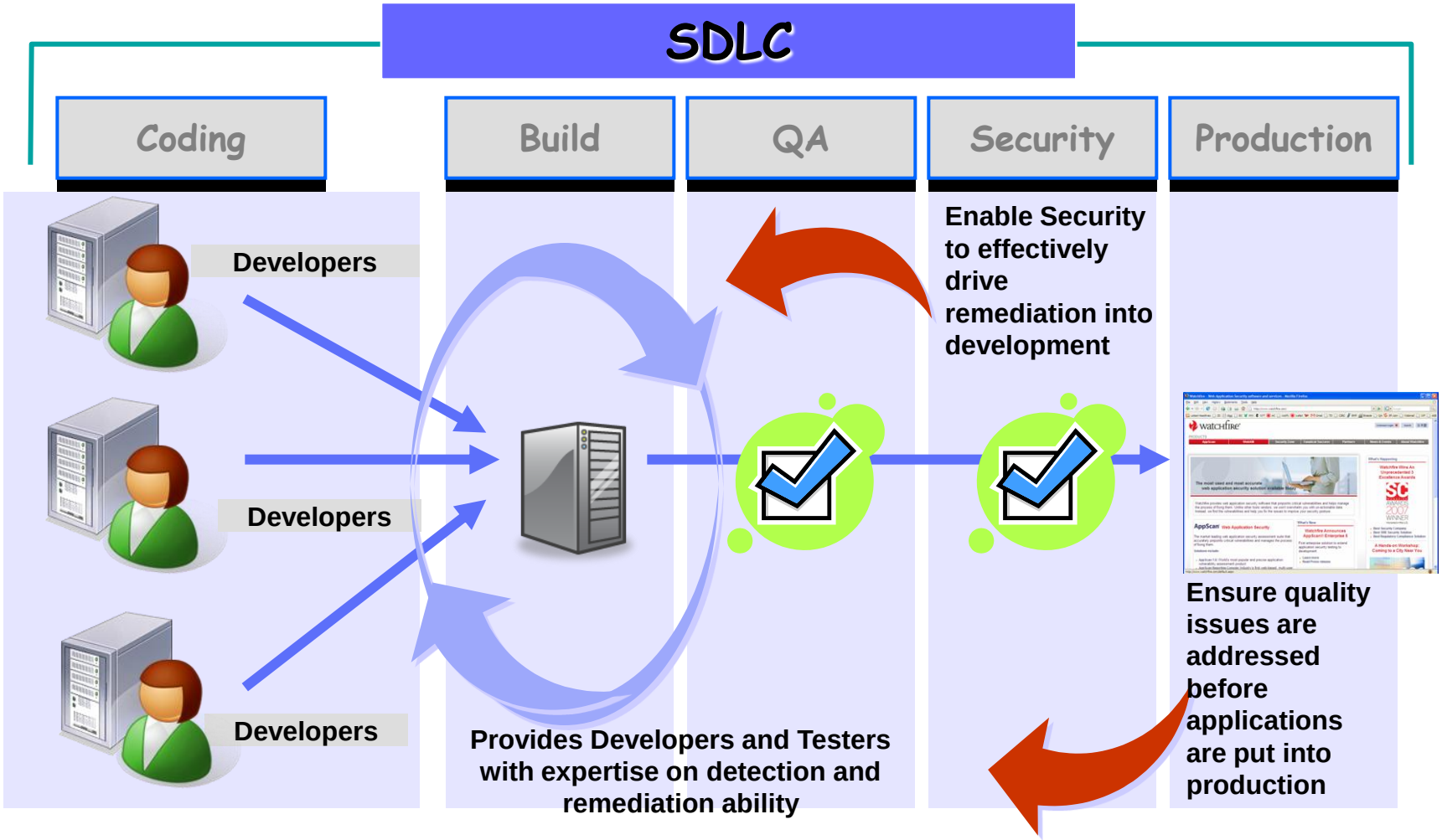
- Automate team workflow improving productivity
- Automate data collection eliminating administrative overhead
- Real time reporting and alerts reduces project risk

Day One Productivity

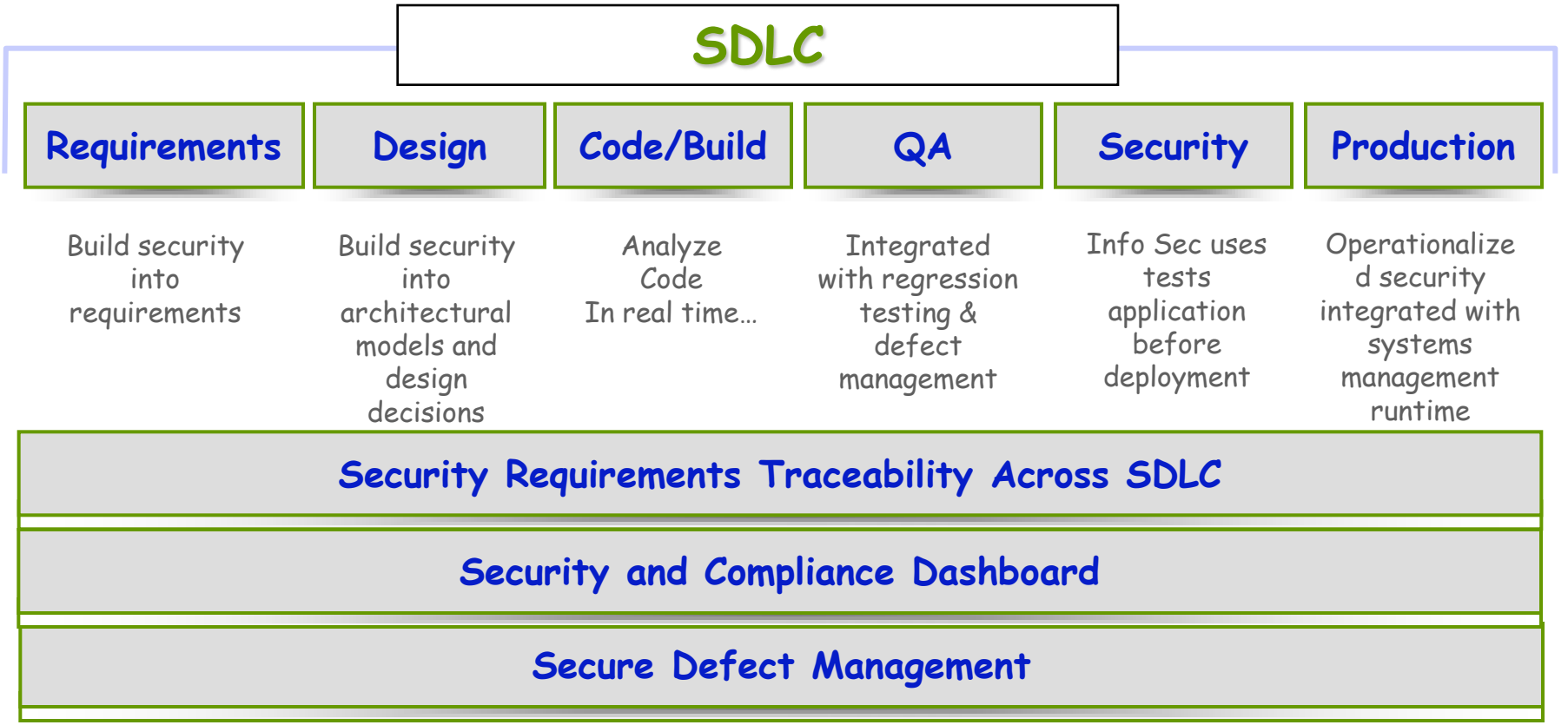
- Dynamic provisioning of projects and teams
- Real-time iteration planning and workload balancing
- Unify teams with tools choice

Real-time integration of people, process and projects across the lifecycle

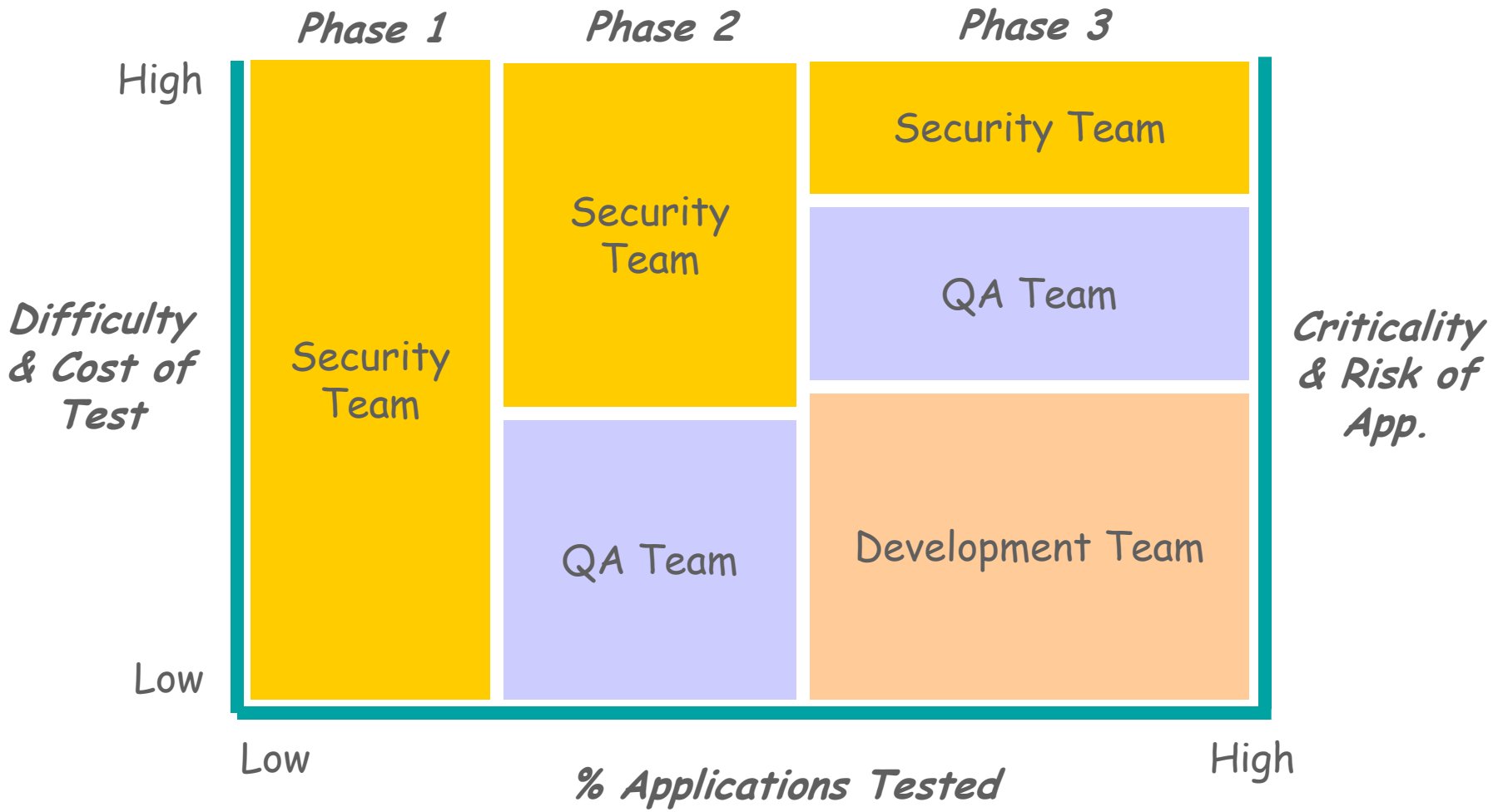
Building security & compliance into the SDLC



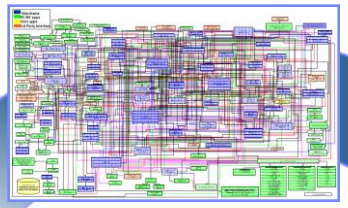
End to End Product Support Across the SDLC



Application Security Adoption Within the SDLC

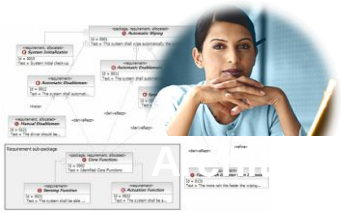


Security compliance = Day-to-Day Lifecycle Management:



Visibility

What's happening with the infrastructure?



Secure Construction

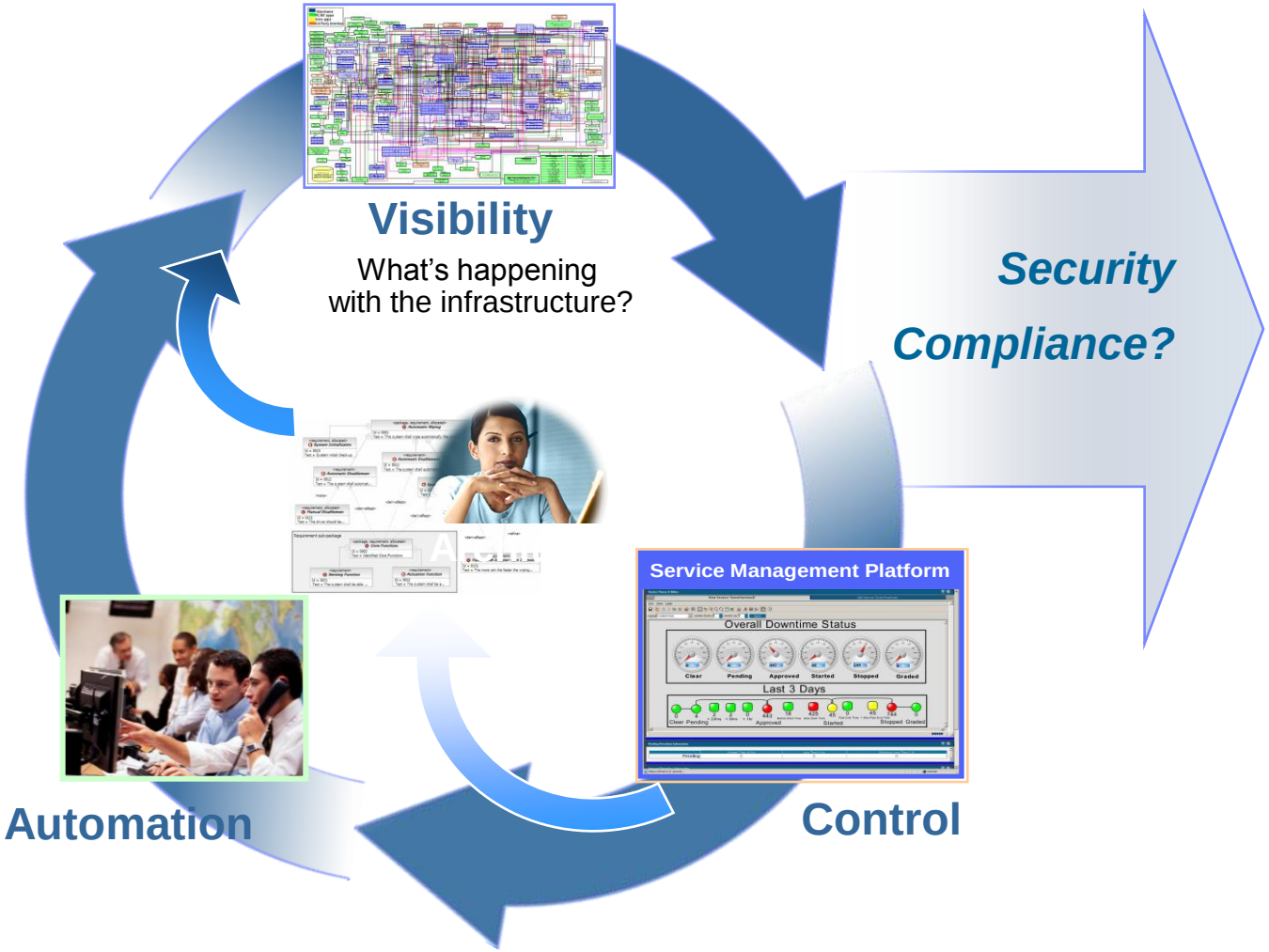
- Secure Coding Method
- Security Architecture Patterns
- Security Analysis Tooling
- Security Quality Test Tools

Security Compliance?

Compliance View:

- Policy Violations
- Vulnerabilities
- Unauthorized access
- Attacks/Responses
- Unauthorized change
- Incident reporting
- Forensics

Security Compliance = Day-to-Day Lifecycle Management:



Compliance View:

- Policy Violations
- Vulnerabilities
- Unauthorized access
- Attacks/Responses
- Unauthorized change
- Incident reporting
- Forensics